CLAIMS

What is claimed is:

1. A method comprising:

assigning a first register class to at least one symbolic register in at least one instruction;

determining and assigning a second register class to the at least one symbolic register;

reducing register class fixups; and

renaming the at least one symbolic register.

2. The method of claim 1, said assigning the first register class is an initial assignment.

3. The method of claim 1, said determining and assigning the second register includes:

marking a register class assignment map at a block entry;

marking a register class assignment map at a block exit;

determining a register class assignment map at an entry of each instruction; and

determining a register class assignment map at an exit of each instruction.

4. The method of claim 1, said reducing register class fixups includes:

hoisting register class fixups;

sinking register class fixups; and

removing unnecessary register class fixups.

5. The method of claim 4, said removing unnecessary register class fixups includes removing dead code.

6. An apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:

assigning a first register class to at least one symbolic register in at least one instruction;

determining and assigning a second register class to the at least one symbolic register;

reducing register class fixups; and

renaming the at least one symbolic register.

7. The apparatus of claim 6, said assigning the first register class instruction is an initial assignment.

8. The apparatus of claim 6, said determining and assigning the second register further including instructions which, when executed by a machine, cause the machine to perform operations including:

marking a register class assignment map at a block entry;

marking a register class assignment map at a block exit;

determining a register class assignment map at an entry of a instruction; and

determining a register class assignment map at an exit of a instruction.

9. The apparatus of claim 6, said reducing register class fixups further including instructions which, when executed by a machine, cause the machine to perform operations including:

hoisting register class fixups;

sinking register class fixups; and

removing unnecessary register class fixups.

10. The apparatus of claim 9, said removing unnecessary register class fixups further including instructions which, when executed by a machine, cause the machine to perform operations including removing dead code.

11. A system comprising:

a processor having at least one register; and

a compiler coupled to the processor executing in a host device that inputs a source program having a plurality of operation blocks,

wherein the compiler assigns a first register class in at least one instruction to the at least one symbolic register, determines and assigns a second register class to the at least one symbolic register, reduces register class fixups, and renames the at least one symbolic register.

12. The system of claim 11, wherein the first register class assigned is an initially assigned register class.

13. The system of claim 11, wherein the second register class determined and assigned includes:

marking a register class assignment map at a block entry;

marking a register class assignment map at a block exit;

determining a register class assignment map at an entry of a instruction; and

determining a register class assignment map at an exit of a instruction.

14. The system of claim 11, said reduction of register class fixups includes:

hoisting register class fixups;

sinking register class fixups; and

removing unnecessary register class fixups.

15. The system of claim 14, said removing unnecessary register class fixups includes removing dead code.

16. A computer comprising:

at least one processor having at least one register coupled to a first memory and a second memory;

at least one user input device coupled to the processor;

a monitor coupled to the processor, and

a compiler executing in the processor that inputs a source program having a plurality of operation blocks,

wherein the compiler assigns a first register class in at least one instruction to the at least one register, determines and assigns a second register

class to the at least one register, reduces register class fixups, and renames the at least one register.

17. The computer of claim 16, wherein the first register class assigned is an initially assigned register class.

18. The computer of claim 16, wherein the second register class determined and assigned includes:

marking a register class assignment map at a block entry;

marking a register class assignment map at a block exit;

determining a register class assignment map at an entry of a instruction; and

determining a register class assignment map at an exit of a instruction.

19. The computer of claim 16, said reduction of register class fixups includes:

hoisting register class fixups;

sinking register class fixups; and

removing unnecessary register class fixups.

20. The system of claim 19, said removing unnecessary register class fixups includes removing dead code.